



EasyLights

Written By: Steven Robert Cypherd

SUMMARY

[raw]I was working on my Picaxe 28x2 robot from <http://letsmakerobots.com/> and I wanted lights and I needed more sensors. I had four input pins left. With the 28x2 you can make the pins anything that you need. I went back to electronics to find some answers. I looked up shift registers on <http://www.jameco.com/> and I bought the latest CD74HC164 serial-in parallel-out. This is why I made Easy Lights. For two pins you get eight lights. Yes, there is a little flicker. It is a shift register. Each time the lights change you have to shift in all eight lights. On slower systems you can use the Master Reset to clear the lights, but it still flickers when loading the lights.

For the files please see <http://letsmakerobots.com> and <http://instructables.com> as I complete this article.

My programs are just a demonstration to show how my light board works. All you need is something like `doLights` or `shiftout` and maybe `checkLights` to run your lights.

Programming can be fun. I left some mistakes in my Basic Stamp 1 program (`EasyLights1.bs1`) because they are very common mistakes. Look at `tPause` and note that it is a byte. When I changed the timing to a variable I put the pause time of 3000 into `tPause` and I got gunk out. Why? A byte holds a maximum of 255 in decimal terms. Processors like the Basic Stamps use 16 bits or "word" size for all math operations. When you put 3000 into a byte-size variable you only get the low byte of the word size of 3000. That is 184 decimal or 10111000 in binary. The system tells you nothing for math errors like this. You see something wrong with your program. That is all you get. This is a good example of a very hard error to find. Even I think that I can't be that stupid.

`doLights` is an example of programming for a device. A shift register like the CD74HC164 has three inputs (Clock, Data and Master Reset) that control the outputs, which are connected to lights in this case. You set Data to high or low and raise your Clock high, wait and set your Clock low and repeat this for all 8 bits in the register. To clear the lights you can load all 8 lights with Data low or bring Master Reset low, wait and set it high. Master Reset needs to be high for normal operation. Most processors like mine are too slow to necessitate waiting between setting the Clock high and then setting it low.

`checkLights` is my way of efficiently using the bits of each of my lights for easy logical testing. Bits only have two values: 0 for off and 1 for on. `checkLights` sets `litTmp` to the bit value of each light. This makes testing easy. If `litTmp = 1` then `doLitOn`. Simple. Most processors give you access to the bits of the first few bytes of memory. If you tested bit 4 of the byte `lights` (light #5) you would get 16, the position value of that bit in the byte. Messy, to say the least. Logic operators like `and` and `or` will give you the bit position value too. In `Larson` to set the high bit of lights I set `lights` to 128 (the value of bit 7, the high bit in `lights`). Binary notation shows how the processor sees your numbers in binary.

- Position = value
 - Bit 7=128
 - Bit 6=64
 - Bit 5=32
 - Bit 4=16
 - Bit 3=8
 - Bit 2=4
 - Bit 1=2
 - Bit 0=1

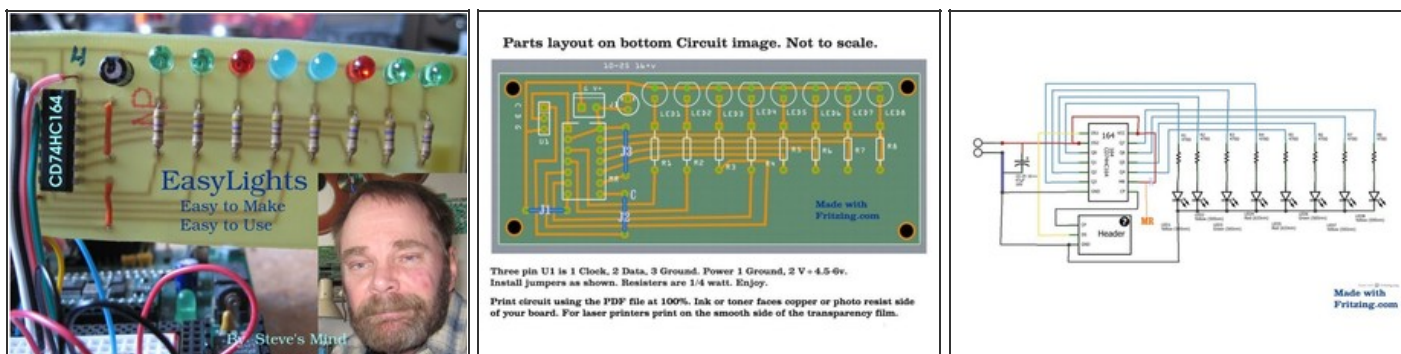
- Examples:
 - Bit 0 + Bit 1 = 3
 - Bit 2 + bit 0 = 5

It is hard being a geek because no one ever understands you. I wish I had friends that I could share this with. I know how lost I can get in my projects. Geeks have hearts too and pretty normal lives.

Easy Lights would be a great project to teach electronics, programming, processors and making boards. If you are not making money on my ideas then have fun. If you make money then contact me, please.

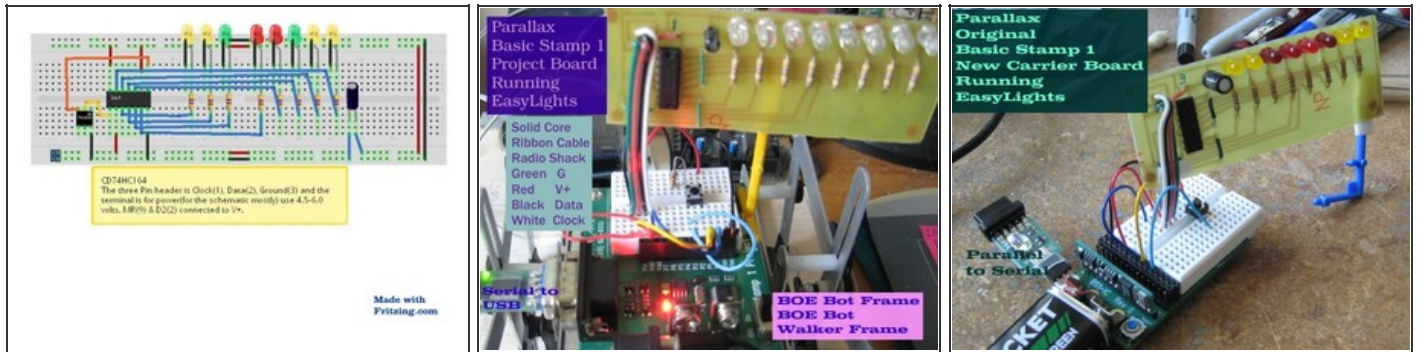
The PDF file contains the circuit board layout.

Step 1 — EasyLights



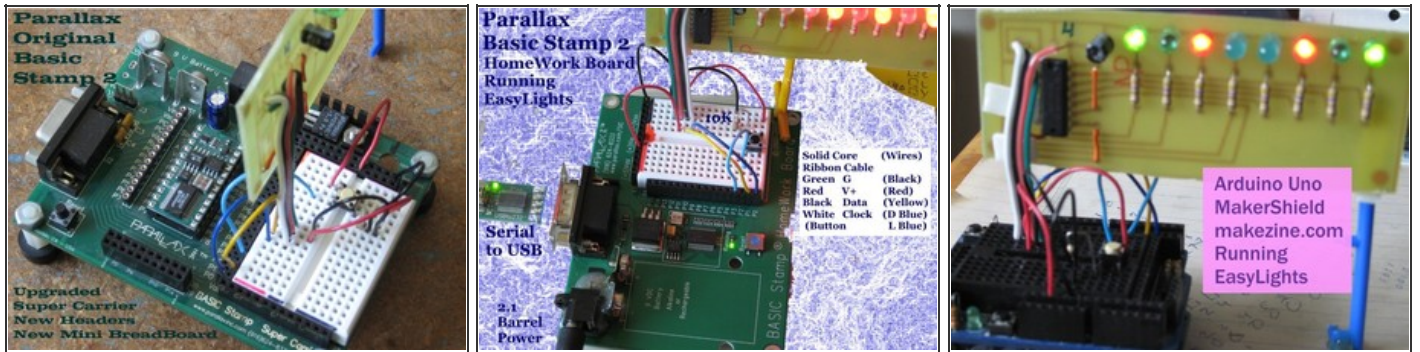
- You can use these processors and programs to run my EasyLights:
- The Basic Stamp 1 version has a button and three modes: Larson, Binary and Random. I have an old original Basic Stamp 1 that I just bought a new carrier board for and a new Basic Stamp 1 project board.
- **EasyLights1.bs1**: Shows a couple of errors. See what you can find.
- **EasyLights2.bs1**: Fixed everything, I think. Used `pause 1000`. Added random numbers.

Step 2



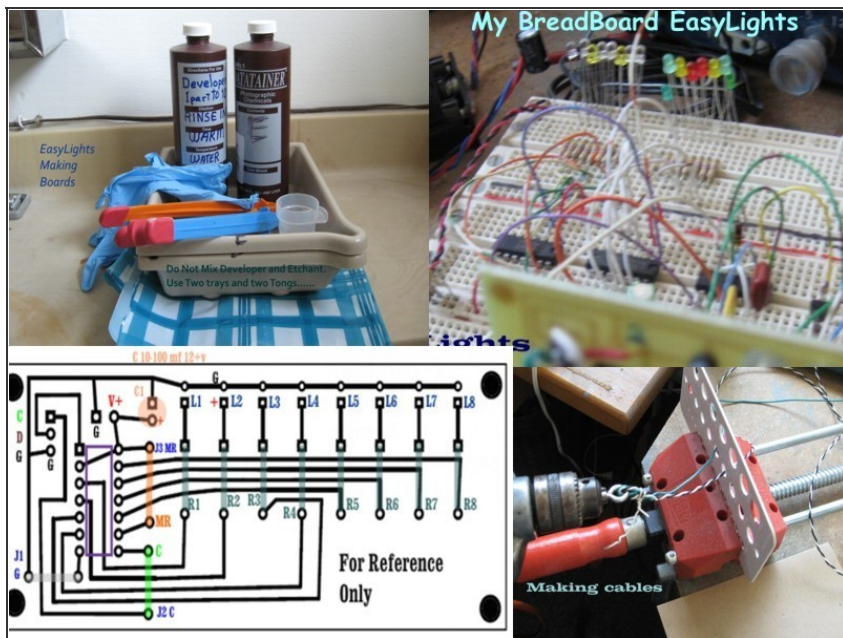
- EasyLights is easy to use and hook up.
- The Basic Stamp 2 version has a button and three modes: Larson, Binary and Random. Plus, I run the "overstuffing a byte" error using the debug terminal. I will try to add a speed button. I have an old original Basic Stamp 2 that I upgraded its super carrier board with new headers and a cute mini breadboard. I also have two Sumo Bot Basic Stamp 2s that I am working on. I also have a Basic stamp 2 project board that is great for learning Basic Stamp 2 programming.
- **EasyLights2.bs2:** Getting started
- **EasyLights3.bs2:** Basics are done: Button, Larson, Binary and Random modes.
- **EasyLights4.bs2:** Added the "overstuffing a byte" error.
- **EasyLights5.bs2:** Added a flag to turn off "overstuffing a byte". Fixed Binary mode.

Step 3



- Running EasyLights:
- The Picaxe 28x2 version has two buttons: mode and speed. Modes are Larson, Binary and Random. This is where Easy Lights comes from. I am making a lights plus analogue board for my robot from <http://letsmakerobots.com>. This is a perfect starter robot for anyone. You get the Picaxe 28 board with an L293 motor driver chip, two right-angle motors with large wheels. a Sharp GP2D12 Infrared range finder and one full-size servo. There are tons of ideas on the web site.
- **easyLights1.bas**: This program runs on the Picaxe.

Step 4



- Last is the Arduino Uno with a Maker Shield and one button running Binary, Larson and Random modes. I got my Arduino at the <http://makezine.com>. I got a full kit with electronic parts, *How to Program Arduino* book and a Maker Shield to build. I love C and that is what you program the Arduinos with. Note: The file extension has changed to .ino with the new Arduino environment. It will still open .pde files but it only saves as .ino files. This is my favorite system.
- **easyLights1.ino**: Getting started
- **easyLights2.ino**: Cleaned it up and added a subroutine for Larson mode. I use the `shiftout` function. Perfect for a shift register. Fun.
- Making boards is so easy now. Last picture lower right is making twisted cables.

Programming is fun!

This document was last generated on 2012-12-24 05:30:30 AM.